Versatile Data Services for Computational Science Applications

Rob Ross
Mathematics and Computer Science Division
Argonne National Laboratory
rross@mcs.anl.gov

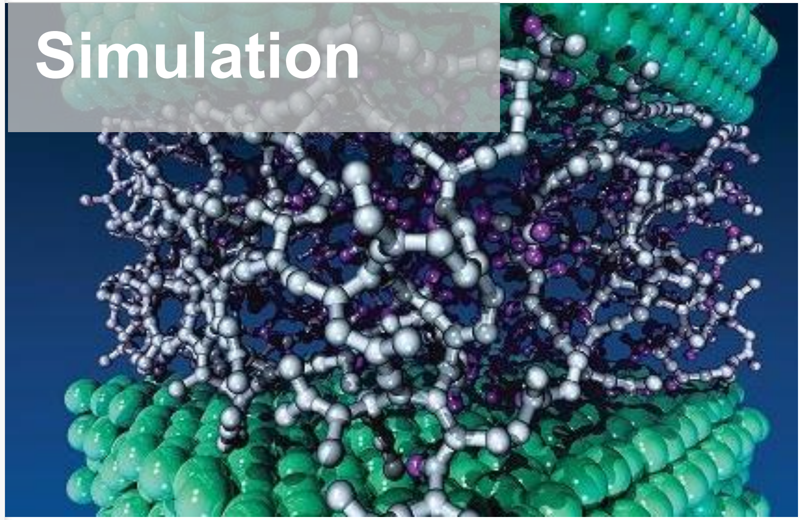
Philip Carns, Matthieu Dorier, Kevin Harms,
Robert Latham, and Shane Snyder
Argonne National Laboratory

Sam Gutierrez, Bob Robey, Brad Settlemyer,
and Galen Shipman
Los Alamos National Laboratory

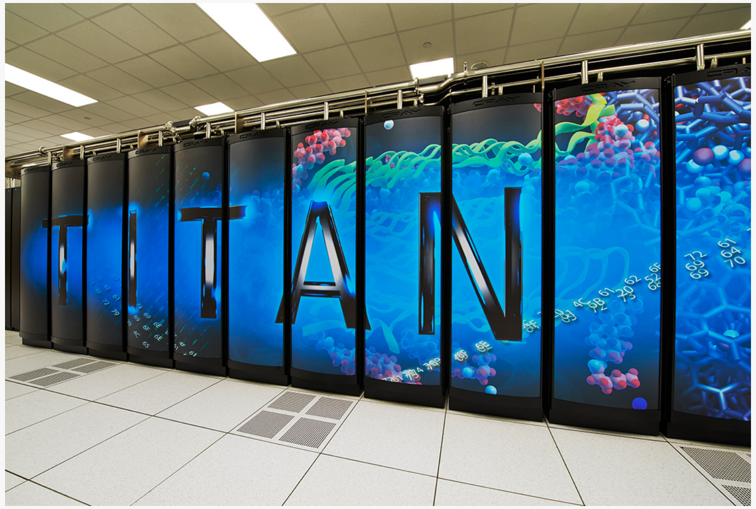
George Amvrosiadis, Chuck Cranor,
Greg Ganger, and Qing Zheng
Carnegie Mellon University

Jerome Soumagne, Neil Fortner
The HDF Group

New Science and Systems: Leading to New Services?

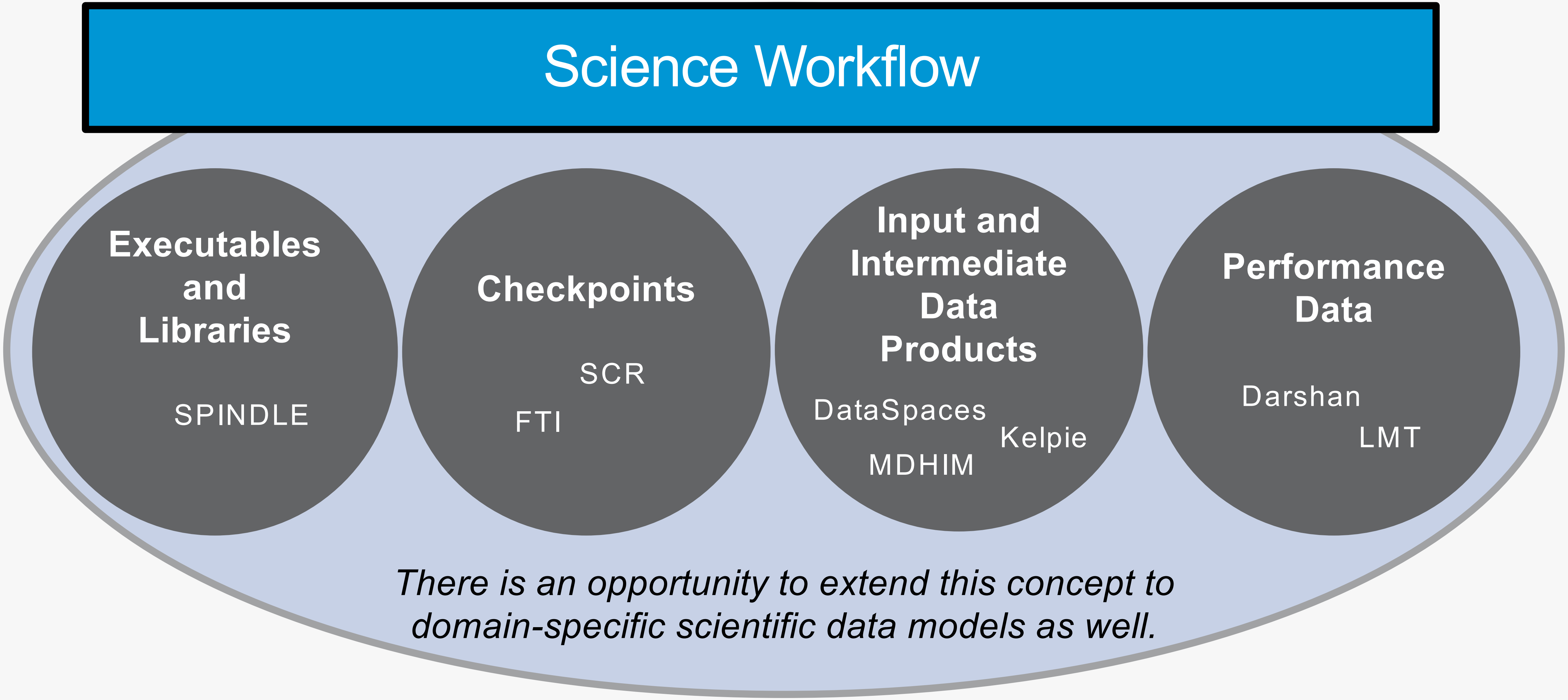


Learning
Growth Prediction Error
0.02
0.00
−0.01
B
mean(BestComboScore)
0
5
10
15
20
25
A
Type
Breast
CNS
Colon
Lung
Leukemia
Melanoma
Ovarian
Prostate
Renal



Top image credit B. Helland (ASCR). Bottom left and right images credit ALCF. Bottom center image credit OLCF.

Data Services in Computational Science



Science Workflow

Executables and Libraries

SPINDLE

Checkpoints

SCR

FTI

Input and Intermediate Data Products

DataSpaces

Kelpie

MDHIM

Performance Data

Darshan

LMT

There is an opportunity to extend this concept to domain-specific scientific data models as well.

Lots of Common Functionality

	Provisioning	Comm.	Local Storage	Fault Mgmt. and Group Membership	Security
ADLB *Data store and pub/sub.*	MPI ranks	MPI	RAM	N/A	N/A
DataSpaces *Data store and pub/sub.*	Indep. job	Dart	RAM (SSD)	Under devel.	N/A
DataWarp *Burst Buffer mgmt.*	Admin./ sched.	DVS/ Inet	XFS, SSD	Ext. monitor	Kernel, Inet
FTI *Checkpoint/restart mgmt.*	MPI ranks	MPI	RAM, SSD	N/A	N/A
Faodel *Dist. in-mem. key/val store*	MPI ranks	Opbox	RAM (Object)	N/A	Obfusc. IDs
SPINDLE *Exec. and library mgmt.*	Launch MON	TCP	RAMdisk	N/A	Shared secret

Reusability in (data) service development.

Productively Developing High-Performance, Scalable (Data) Services

Mochi

Vision

- Specialized data services
- Composed from basic building blocks
- Matching application requirements and available technologies
- Constraining coherence, scalability, security, and reliability to application/workflow scope

Approach

- Lightweight, user-space components and microservices
- Implementations that effectively utilize modern hardware
- Common API for on-node and off-node communication

Impact

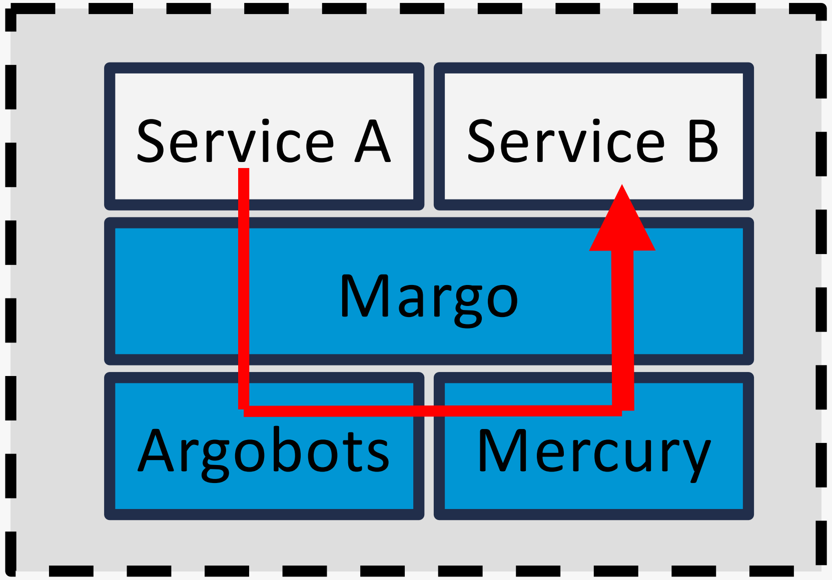
- Better, more capable services for DOE science and facilities
- Significant code reuse
- Ecosystem for service development, float all boats

See http://www.mcs.anl.gov/research/projects/mochi/.

Carnegie Mellon University HDF Los Alamos NATIONAL LABORATORY Argonne NATIONAL LABORATORY

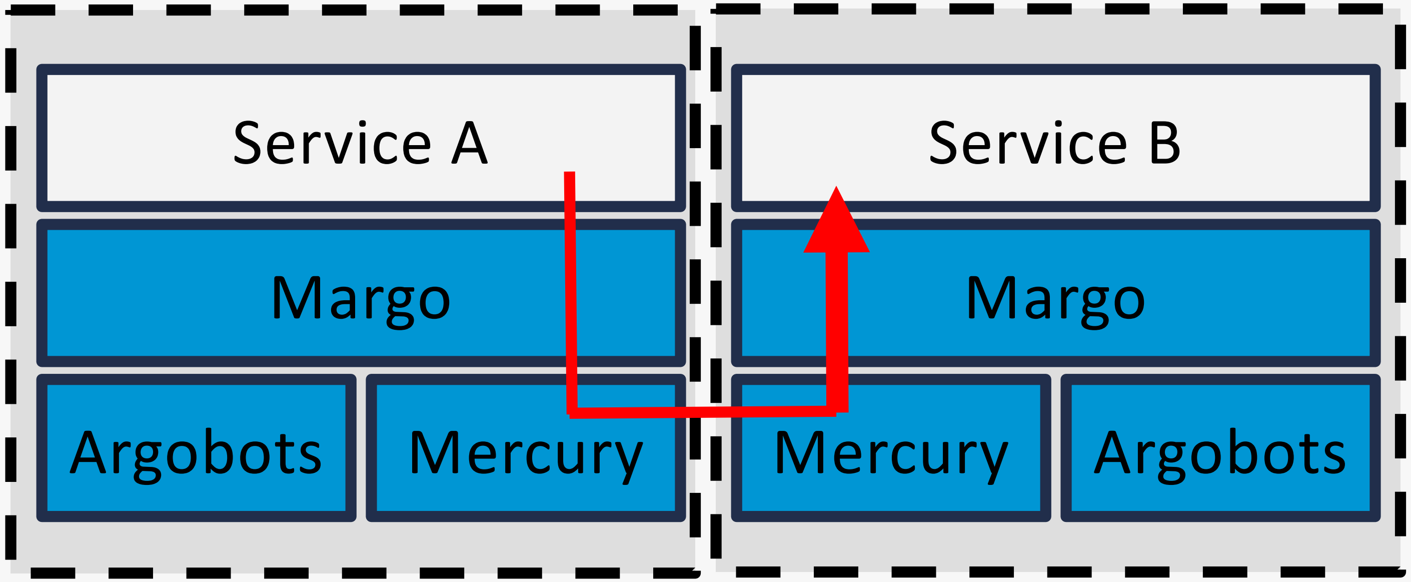
Building Mochi Components

- Mercury: **RPC/RDMA** with support for shared memory and multiple native transports
- Argobots: **Threading/tasking** using user-level threads
- Margo: Hide Mercury and Argobots details, **focus on RPC handlers**
- *Thallium*: **C++14 bindings**



Single Process:

- Direct execution of RPC handlers



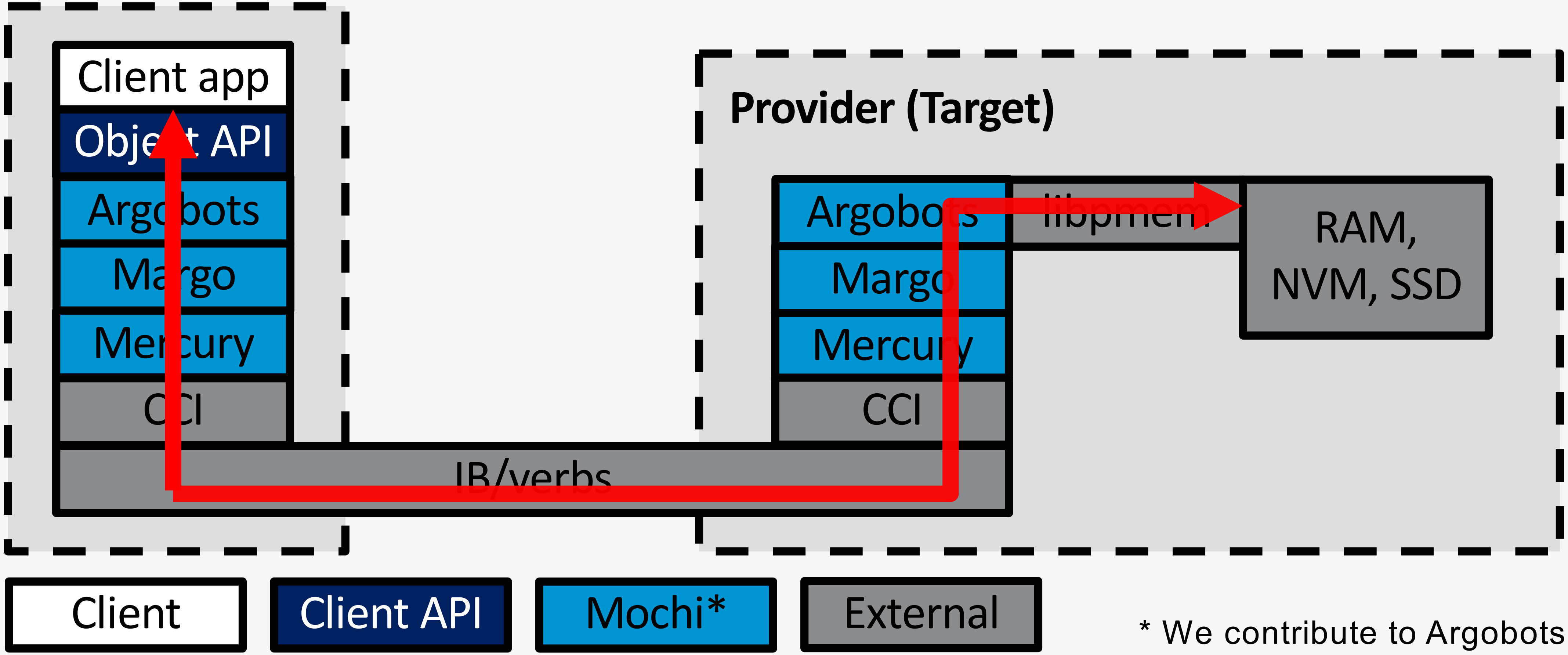
Separate Processes:

- Shared memory (separate processes on same node)
- RPC and RDMA over native transport (separate nodes)

More Components!

- **BAKE**: RDMA-enabled data transfer to remote storage (e.g. SSD, NVRAM)
- **SDS-KeyVal**: Key/Value store backed by LevelDB or BerkeleyDB
- **Scalable Service Groups (SSG)**: group membership management using gossip
- **PLASMA**: Distributed approximate k-NN database
- **POESIE**: Enables running Python and Lua interpreters in Mochi services
- **Python wrappers**: Py-Margo, Py-Bake, Py-SDSKV, Py-SSG, Py-Mobject, etc.
- **MDCS**: Lightweight diagnostic component

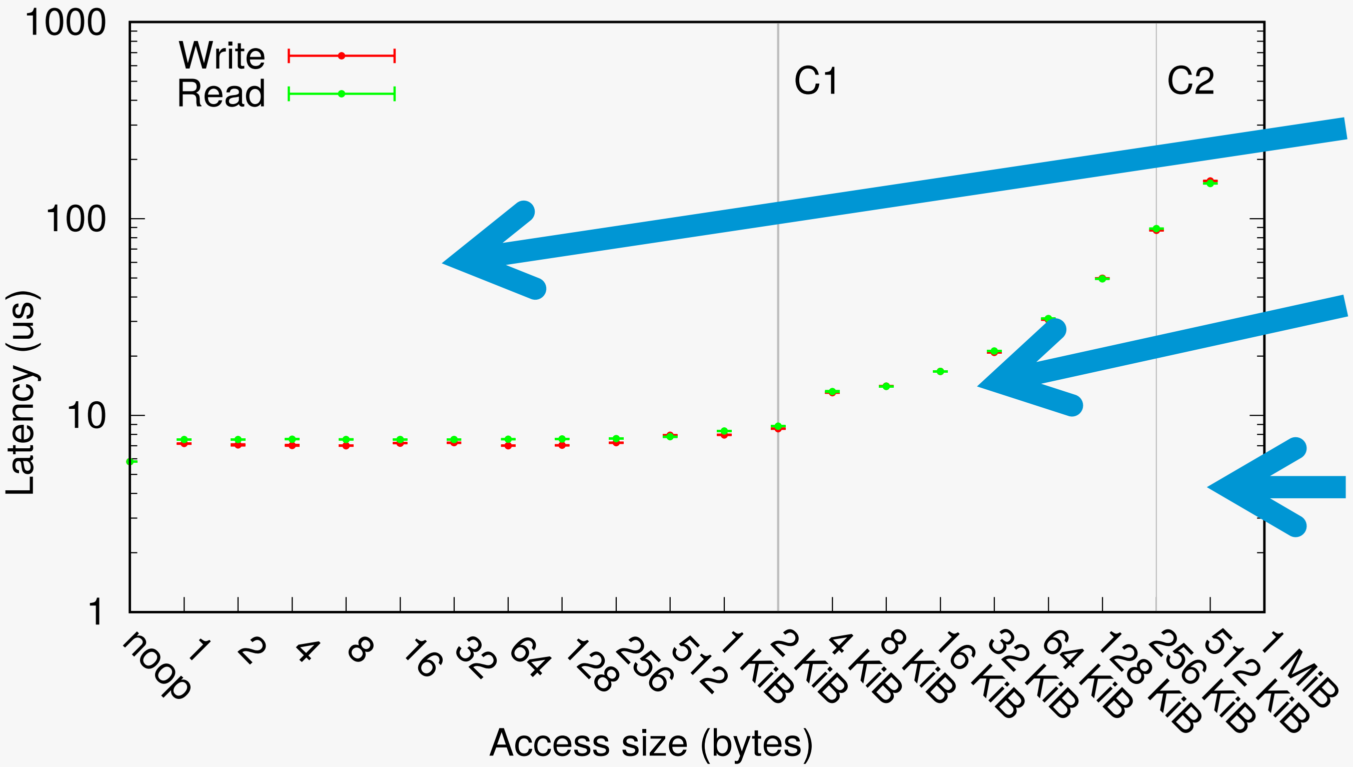
BAKE: A Composed Service for Remotely Accessing Objects



* We contribute to Argobots, but it's primarily supported by P. Balaji's team.

P. Carns et al. "Enabling NVM for Data-Intensive Scientific Services." INFLOW 2016, November 2016.

BAKE: Latency of Access



Multiple protocols:

Small: data is packed into RPC msg

Medium: data is copied to/from pre-registered RDMA buffers

Large: RDMA “in place” by registering memory on demand

- Haswell nodes, FDR IB
- Backing to RAM rather than persistent memory
- No busy polling
- Each access is at least 1 network round trip, 1 libpmem access, and 1 new (Argobots) thread

Examples of composed services.

HEPnOS: Fast Event-Store for High-Energy Physics (HEP)

Goals:

- Manage physics event data from simulation and experiment through multiple phases of analysis
- Accelerate access by retaining data in the system throughout analysis process

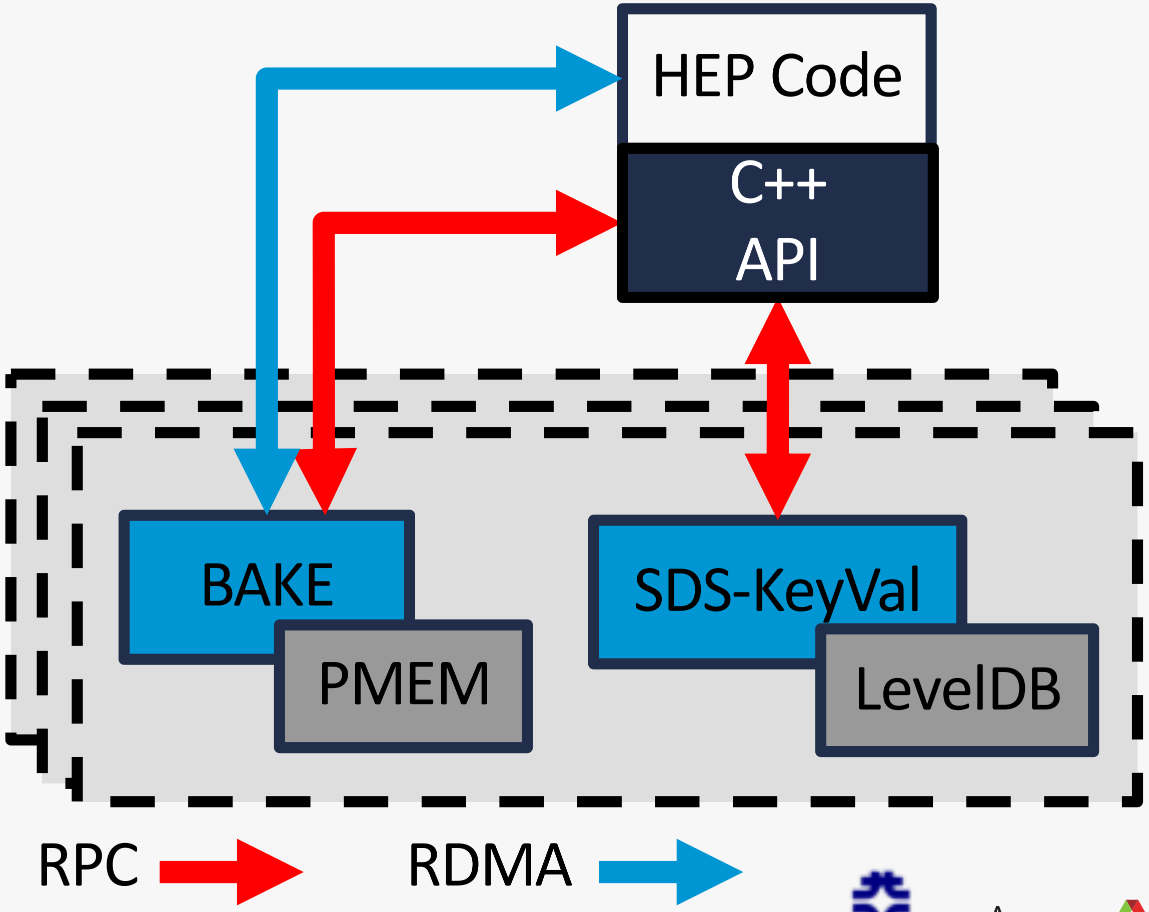
Properties:

- Write-once, read-many
- Hierarchical namespace (datasets, runs, subruns)
- C++ API (serialization of C++ objects)

Components:

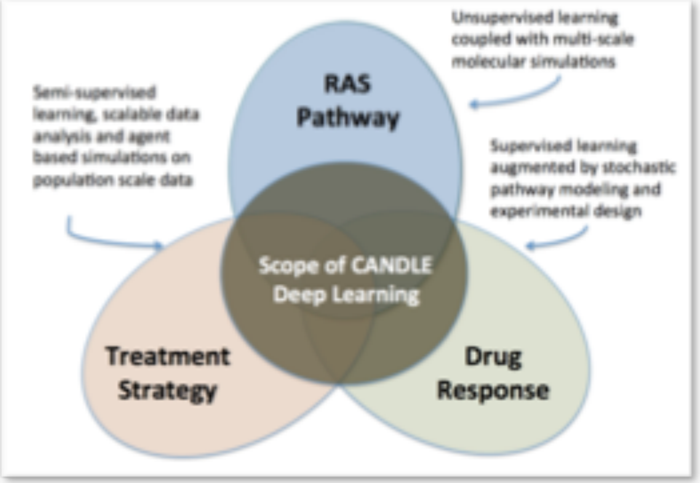
- Mercury, Argobots, Margo, SDSKV, BAKE, SSG
- **New code: C++ event interface**

Map data model into stores



Collaboration with FermiLab led by J. Kowalkowski.

FlameStore: A Transient Storage System for Deep Neural Networks


Unsupervised learning coupled with multi-scale molecular simulations
Semi-supervised learning, scalable data analysis and agent based simulations on population scale data
RAS Pathway
Supervised learning augmented by stochastic pathway modeling and experimental design
Scope of CANDLE Deep Learning
Treatment Strategy
Drug Response

Goals:

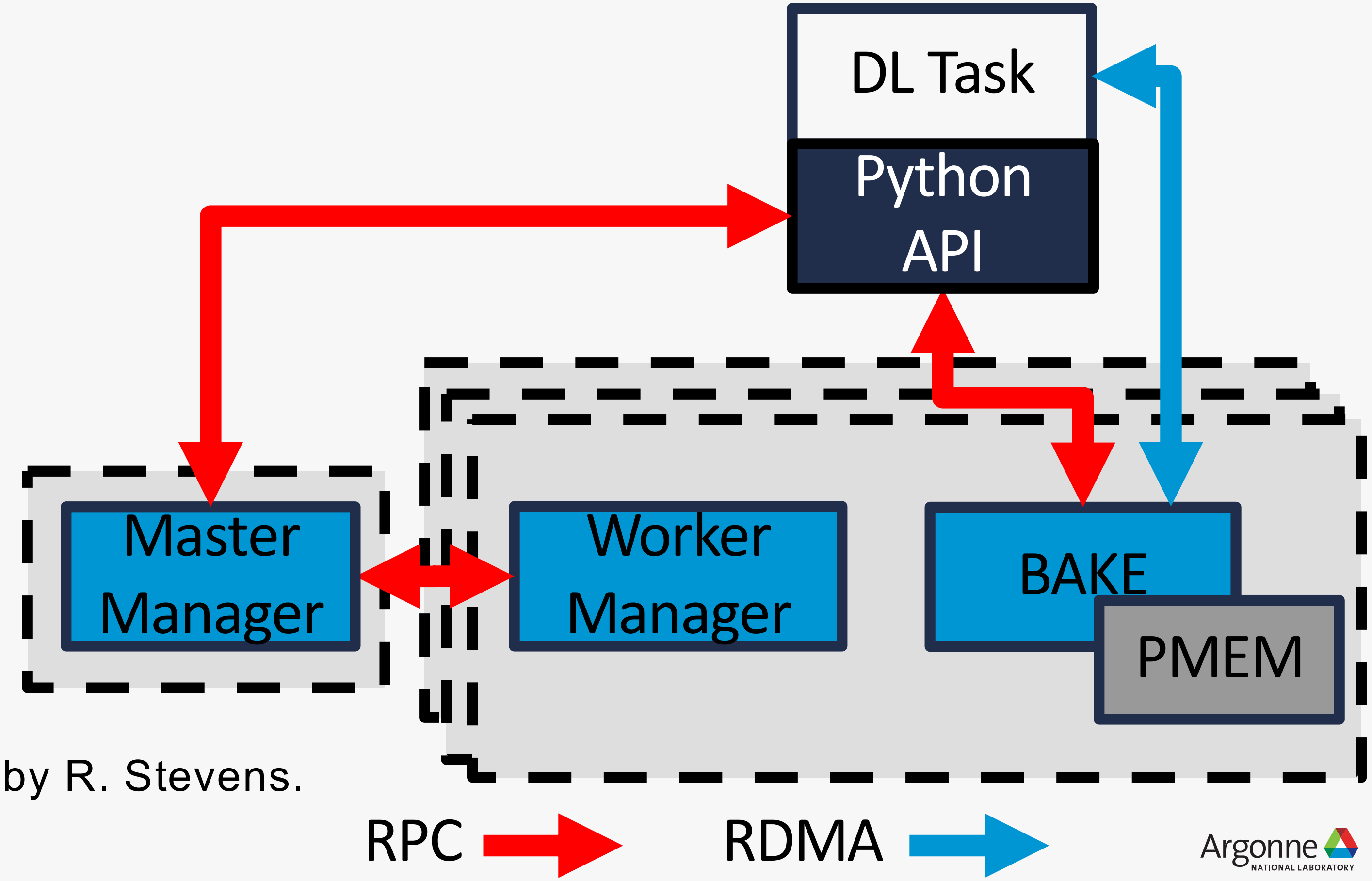
- Store a collection of deep neural network models during a deep learning workflow
- Maintain metadata (e.g., hyperparameters, score) to inform retention over course of workflow

Properties:

- Write-once-read-many
- Flat namespace
- High level of semantics
- Python API (stores Keras models)

Components:

- Mercury, Argobots, Margo, BAKE, POESIE, and their Python wrappers
- **New code: Python API, master and worker managers**



Collaboration with CANDLE cancer project, led by R. Stevens.

Mobject: An Object Store Composed from Microservices

Goals:

- Validate approach with a more complex model
- Provide familiar basis for use by other libraries (e.g., HDF5)

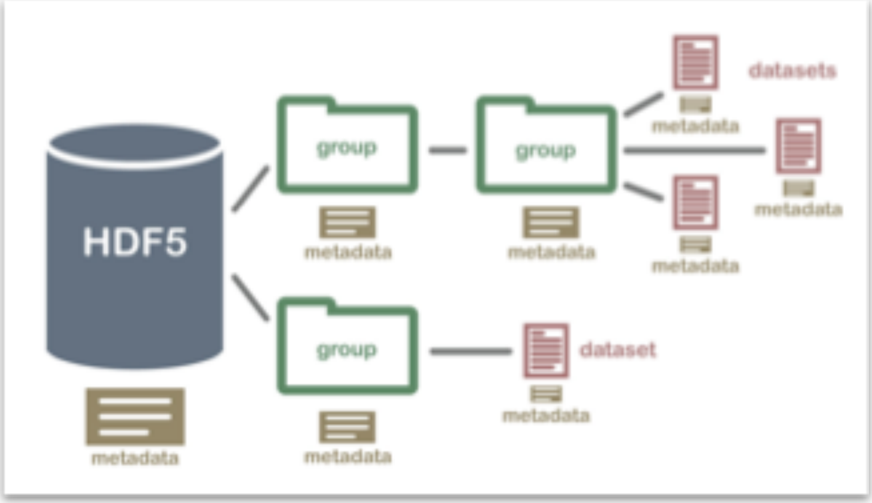
Properties:

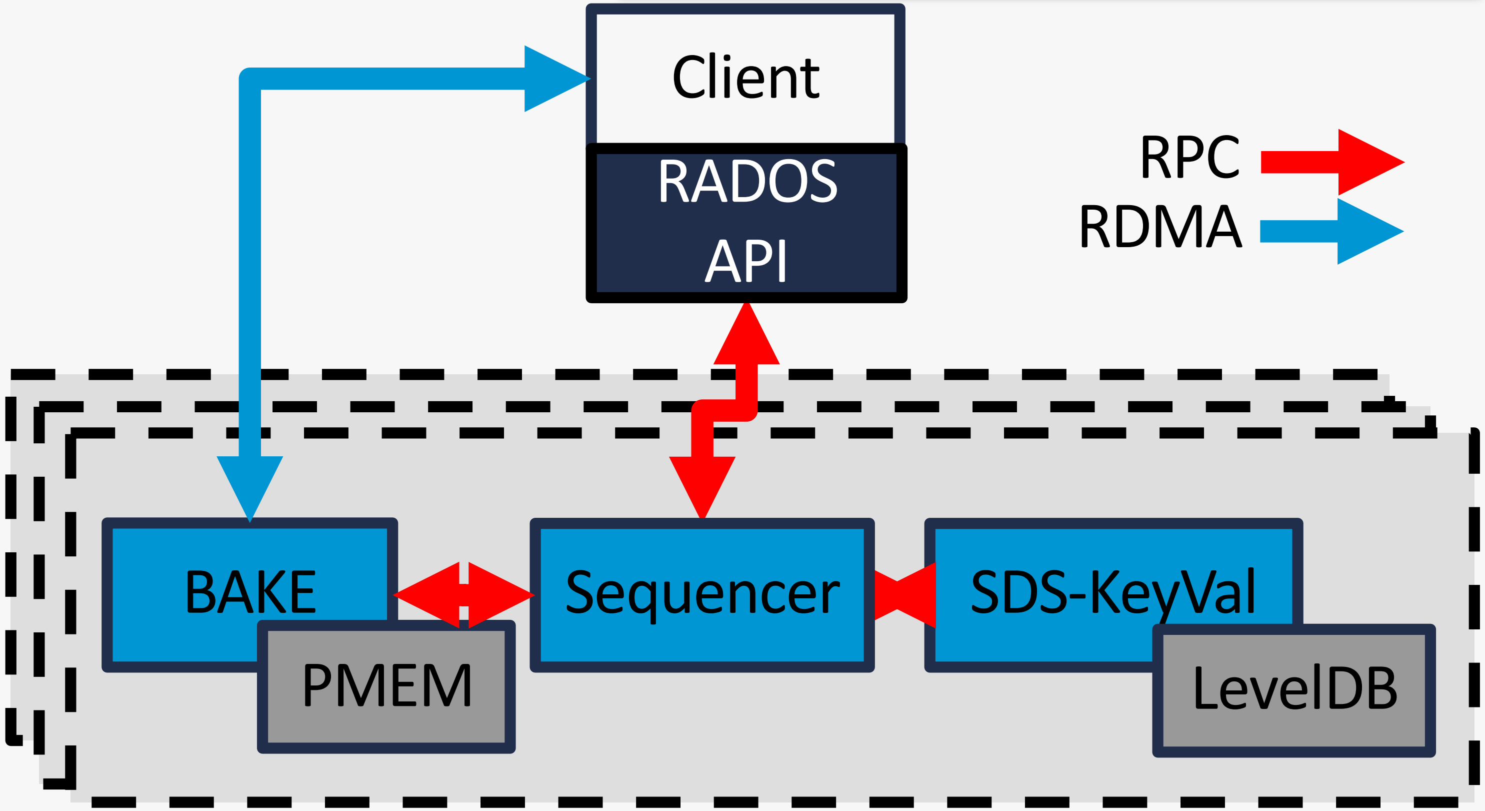
- Concurrent read/write
- Flat namespace
- RADOS client API (subset)

Components:

- Mercury, Argobots, Margo, SDSKV, BAKE, SSG
- **New code: Sequencer, RADOS API**

Collaboration with the HDF Group.





Why am I here?

Learning about this community, but also …

- **How should we analyze these services?**
- **Looking for potential users and collaborators!**
  - Performance data management service?

    Thomas Ilsche et al., “Optimizing I/O forwarding techniques for extreme-scale event tracing”, Cluster Computing Journal, June 2013.
- **Interested in how others build distributed services in HPC**
- **Thinking about autonomics, implementing control loops**
  - Real-time performance analysis
  - Architecture for (decentralized) control of (multi-component) services

Thanks!

This work is in part supported by the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357; in part supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative; and in part supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program.

http://www.mcs.anl.gov/research/projects/mochi/